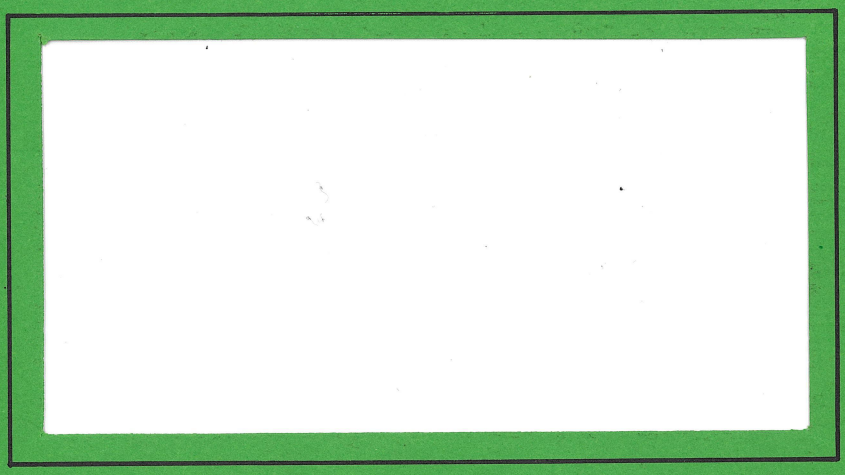

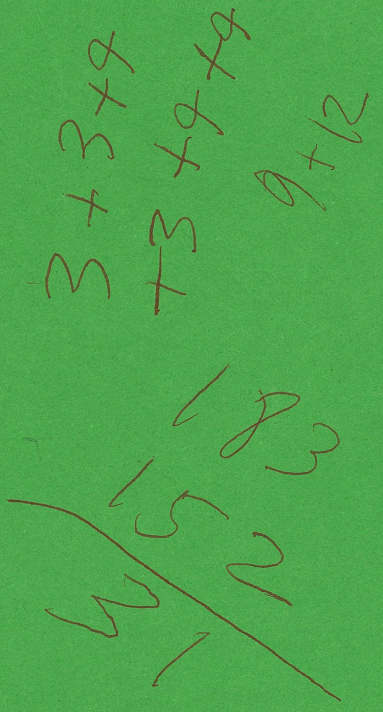

Computer Science
University of Utah

DEPARTMENT OF COMPUTER SCIENCE

HANDBOOK

1980 - 1981

Department of Computer Science
Room 3160 Merrill Engineering Building
University of Utah
Salt Lake City, Utah 84112

Revised October 1980

Table of Contents

1. DEPARTMENT OF COMPUTER SCIENCE FACULTY

Acting Chairman
Laurence H. Lattman

Associate Chairman
Richard F. Riesenfeld

Professors

Robert E. Barnhill, Ph.D.
Anthony C. Hearn, Ph.D.
(on leave 1980-81)
Elliott I. Organick, Ph.D.
Robert E. Stephenson, Ph.D.
William J. Viavant, Ph.D.

Associate Professors

Alan L. Davis, Ph.D.
Martin L. Griss, Ph.D.
Lee A. Hollaar, Ph.D.
Robert M. Keller, Ph.D.
Gary E. Lindstrom, Ph.D.
Suhas S. Patil, Sc.D.
Richard F. Riesenfeld, Ph.D.

Assistant Professors

Steven F. Boll, Ph.D.
P. A. Subrahmanyam, Ph.D.

Research Associate Professors

Ercolino Ferretti, B.M.
Kent F. Smith, M.S.

Research Assistant Professors

Brent S. Baxter, Ph.D.
Elaine Cohen, Ph.D.

Adjunct Professors

David C. Evans, Ph.D.
Thomas G. Stockham, Jr. Sc.D.

2. DEPARTMENT OVERVIEW

The Department undergraduate program offers study of Computer Science leading to the B.S. degree. The undergraduate curriculum provides a general education in mathematics, science and humanities; an in-depth study of information processing systems, including the practical and theoretical aspects of hardware and software; specialized study in selected areas of Computer Science; and introductions to the use of computing systems as tools for students of many backgrounds and interests. Complete requirements for the B.S. degree are discussed in the Undergraduate Information section of this handbook.

Graduate degrees offered by the Department are the M.S., Ph.D., M.E., and M.Phil. The graduate program is open to Computer Science and Mathematics graduates and also to students whose preparation is outside of Computer Science. However, students entering graduate study from outside the Computer Science or Mathematics areas may require additional preparation before beginning their graduate program.

The Department graduate program offers students the opportunity to involve themselves in any of several areas of research specialization including: asynchronous computation, computer aided geometric design, computer systems, software for small computer systems, parallel computation, programming languages, program verification, sensory information processing, very large scale integrated circuit design, symbolic and algebraic computation, theory of computation, and several applications areas.

FACILITIES

The central Departmental computing facility is a Digital Equipment Corporation DECSystem 2060, which supports all research projects and graduate level classes.

Additional Departmental computing facilities used to support various research efforts and classes include a Burroughs B1865, a DEC PDP-10 which runs in a single user mode, a DEC PDP-11/45 with an FPS-120B array processor, and a graphics laboratory consisting of a DEC PDP-11/60, an Evans & Sutherland Picture System I (to be replaced with an Evans and Sutherland Multi Picture System in the fall of 1980) and a Grinnell color frame buffer. The Department also operates a Small Computer Laboratory consisting of a PDP 11/34 supporting software development on numerous micro-computer systems. Other new facilities to be installed in the fall of 1980 include a Hewlett Packard System 3000/33, a ComputerVision computer aided design system, and an on-line photocomposition system for use with the DECSystem 20.

It is also possible to design and fabricate LSI circuits using the

facilities of the VLSI Laboratory in the Department and the HEDCO Microelectronics Laboratory in the same building. These facilities include a CAD machine for designing LSI circuits and maskmaking, fabrication, and testing facilities. The Department of Computer Science also maintains special research facilities for sensory information processing, photographic processing and small scale digital system assembly and study. The University Computer Center houses a Univac 1100/60 which is also used by the Department primarily for introductory and service courses.

The laboratories that house these facilities are summarized below:

SOFTWARE RESEARCH LABORATORY. Located here are a Burroughs B1865 computer and a Digital Equipment Corporation DECSystem 2060. The B1865 is a microprogrammable machine whose flexible architecture makes possible the simulated design, construction, testing and monitoring of other computer systems by emulation. Implementation of a microcoded LISP system is an example of one project currently using the capabilities of this system.

The DEC 2060 has over two megabytes of memory and supports research projects in the Department by providing a general purpose time-sharing environment with over 70 terminals throughout the Department. It also has the usual complement of disk, tape and other I/O devices including a four color plotter, several office-quality (daisy wheel) printers, and links to most other departmental facilities. Most graduate student and faculty offices have terminals connected to the DECSystem 2060, and additional terminals are available in several terminal workrooms. The planned addition of an extremely high quality photocomposition device on-line to the DECSystem 2060 in the fall of 1980, in conjunction with several excellent software systems, provide a very attractive facility for document composition and preparation. This facility, combined with an operating system (TOPS 20) on the DECSystem 2060 known for its extremely effective and "friendly" human interface, provides a computing environment in which users can work with maximum efficiency and enjoyment.

The 2060 also supports a highly sophisticated electronic mail system, which handles virtually all intra-departmental communication. The 2060 is also connected to both the ARPANET and TELENET networks, enabling remote access to its resources and the sharing of resources with other computer systems on these networks. A local computing network is under development, and it is currently possible to connect to all other systems in the Department from the DEC 2060.

SMALL COMPUTER LABORATORY. The Small Computer Laboratory was established as a center of focus for Departmental activities dealing with software for small computers. This laboratory houses a medium scale DEC PDP 11/34 running UNIX, which in turn supports numerous 8 and 16 bit micro computers, by providing cross compilation and support services (line printer, tape drive, large disk system). The micro computer systems are used for experiments in software

portability and networking of small computer systems, as well as other projects. They are available for students who need "hands on" access for individual or group programming projects and are often used to support classes in software engineering and systems programming. In the fall of 1980 this laboratory will acquire a Hewlett Packard System 3000/33, for use primarily in support of research and instruction in integrated circuit design. Unused time on this system will be made available for other uses.

DIGITAL SYSTEMS LABORATORY. The Digital Systems Laboratory provides an environment in which students having no prior experience with electronic circuitry can safely and quickly build digital and hybrid computing devices from the most primitive decoding nets up to small but functional digital computers with graphics output. An open shop projects course is taught in coordination with the sophomore course in computer fundamentals. Advanced students are encouraged to attempt projects involving both hardware and software. A PDP 11 minicomputer is available for this purpose. The Laboratory is also available for senior projects, independent study, and graduate research projects.

COMPUTER GRAPHICS LABORATORY. This facility contains a state of the art black and white and color picture processing facility. The main computer in this lab is a DEC PDP 11/60 running UNIX, and is connected to the Department's DEC 2060 by a shared disk drive. Attached to the PDP 11/60 is an Evans and Sutherland Picture System I for black and white line drawings and a Grinnell color picture processing system and frame buffer for color picture production. The E&S Picture System I is being replaced in the fall of 1980 by an E&S Multi Picture System. Images created in the laboratory can be captured on hard copy either through use of the Department's Photographic Laboratory facilities, or on devices such as the four color plotter. The laboratory is also acquiring a ComputerVision computer aided design system for use in integrated circuit design and general computer graphics.

SENSORY INFORMATION PROCESSING LABORATORY. This laboratory has two large on-site computers; a DEC PDP-10 and a DEC PDP-11/45. The former is a "dedicated" processor available to a single user, while the latter supports one foreground user and one or two background users. These computers are used for research work in audio or picture processing, much of which is of a real time nature. They drive an assortment of visual and audio I/O equipment, the latter in the "Quiet Room." The Laboratory has extensive communication equipment to connect the systems to each other and remote terminals. Facilities of additional remote computers are available over the ARPA network.

COMPUTER SCIENCE LIBRARY. This library is located in Room 3147 MEB and is available to all Computer Science students. It contains many texts and monographs from the Computer Science field as well as current journals and technical reports of research activities.

PHOTOGRAPHIC LABORATORY. The Department operates a photographic laboratory capable of producing extremely high quality black and white and color images. This laboratory is used primarily to support sponsored research activities such as the computer graphics, computer aided design, and VLSI efforts. This laboratory is also used for sponsored research support such as the production of transparencies to accompany oral presentations and photographs for inclusion in technical reports and dissertations.

HEDCO MICROCIRCUITS LABORATORY. The College of Engineering operates, with significant participation from several Computer Science faculty members, a research oriented facility for the fabrication of integrated circuits. Typically the design of integrated circuits is performed using facilities such as the DECSystem 2060 and computer graphics laboratory, and then taken to the Microcircuits Laboratory for fabrication. The Microcircuits Laboratory includes facilities for IC maskmaking and low volume production and testing of IC wafers. This laboratory also has instructional facilities for teaching LSI fabrication techniques.

UNIVERSITY OF UTAH COMPUTER CENTER. This facility consists of a UNIVAC 1100/60 with 768K words of main memory and other large scale on-line storage devices. This facility is used mainly for introductory and service courses taught by the Department. Terminals throughout the campus are connected to this facility by the Department.

3. INFORMATION FOR UNDERGRADUATES

3.1 DECLARING COMPUTER SCIENCE AS A MAJOR

The requirements for a bachelor's degree in Computer Science have a tendency to change. It is therefore advisable to declare Computer Science as your major as soon as you have decided. This will ensure that you are free to follow through with the set of requirements which were in effect when you began your course work. Early declaration of your major will also enable you to participate in Department affairs such as SAC (Student Advisory Committee). The requirements listed in this handbook are in effect for those entering the program in July 1980 or later.

3.2 UNDERGRADUATE ADVISORS

Each student majoring in Computer Science is assigned a faculty undergraduate advisor. Students are asked to meet with the advisor at least once each quarter to discuss registration plans for the next quarter and to discuss current problems, if any, as well as to bring the student up to date on departmental actions that may affect the student's plans. The responsibility for doing this is left to the student.

Questions regarding transfer of credits, degree requirements, course content, etc., should be directed to the advisor. Students should always feel free to seek advice from their advisor regarding their program and plans.

New students should contact the undergraduate secretary in 3162 MEB for assignment to an advisor.

3.3 REQUIREMENTS FOR THE B.S. IN COMPUTER SCIENCE:

1. Overall requirement: Students must meet the University requirement of a minimum of 183 quarter hours of coursework. The Department requires that students maintain accumulative GPA of 2.50 or higher to continue as CS majors, and to graduate. Pre-Majors: Students, including transfer students, not meeting departmental admission requirements will be placed in a pre-major category by the Registrar. Such students are urged to visit the Department Undergraduate Secretary for additional information. Admission to the Department: Upon application to the Department, students will be admitted as fully matriculated Computer Science majors provided they meet the following requirements:

   (a) A cumulative GPA of 2.50 or higher

(b) Completion of the following core courses with a grade of C or better in each course:

Math 111, 112, CS 105; Physics 171, CS/Math 334

Continuing performance: Students are expected to continue to maintain a cumulative GPA of 2.50 to remain in good standing. Each course taken to satisfy requirements 3 through 6 below must be passed with grade of C or better. A student may repeat a course at the 400 level or above only one time. Probation: If a student's overall GPA falls below 2.50, he will be notified that he/she is on probation, and will be given conditions for return to good standing. Normally these conditions must be satisfied during the next two academic quarters. Students failing to meet their probationary conditions will be dropped from Department rolls.

2. Liberal Education Requirement. Completion of the liberal education requirements as described in the current General Catalog and in the Liberal Education Bulletin, and English 301. English 301 must be passed with a grade of "C" or better.

3. CS Core Requirement. A minimum of 51 credit hours of Computer Science courses which must include: 105, 306, 321, 322, 323, 411, 412, 417, 427, and either 418 and 519 or 428 and 529. These are supplemented by at least nine credit hours of courses selected from the current approved elective list (see section below).

4. Math Requirement. Math 111, 112, 113, CS/Math 334, Math 351, Math 352, Math 507, Math 560.

5. Physics and Chemistry requirements: Phy 171, Phy 172, Phy 173; Chem 121.

6. At least 20 hours of approved study must be completed in a single discipline other than Computer Science. This discipline may be freely selected by the student and typically might be in a field where the student is interested in applying his computer science knowledge. Approval for courses selected to satisfy this requirement, however, must be obtained from the student's advisor. For this reason a student should consult with his advisor regarding this requirement as soon as he decides to major in Computer Science.

7. No single course may be counted toward satisfying more than one of the requirements 3, 4, 5, and 6 as outlined above, except in the case of the Physics and Chemistry requirements which may fulfill the Liberal Education requirement for science distribution courses.

3.4 APPROVED LIST OF COURSES FOR THE CS ELECTIVE REQUIREMENT

CS 410 (4) Low Level Computer Programming
CS 490 (1-3) Senior Project
CS 521 (also EE 576) (3) Logical Design*
CS 522 (2) Logic Design Laboratory*
CS 523 (3) Program Verification
CS 524 (4) Switching Circuit Theory
CS 418 (4) and CS 519 (4) Programming Laboratory
CS 428 (4) and CS 529 (3) Digital Systems Laboratory
CS 534 (3) Intro. Theoretical Comp. Sci.
CS 536 (4) Operating Systems
CS 539 (also EE 560) (4) Fund. of Integrated Circuits
CS 540 (also EE 561) (4) Modelling of Integ. Circuits
CS 541 (also EE 572) (3) Simulation of Physical Systems
CS 542 (also EE 577 and IE 542) (3) Simulation of Dis. State Sys.
CS 543 (3) Advanced Dis. State Sys. Sim.
CS 551 (3) Computer Graphics
CS 565 (2) Computer Music Seminar
CS 566 (3) Computing with Symb. Expr.
CS 590 (Arr) Independent Study
EE 321 (5) Electric Circuits
EE 330 (5) Engineering Electronics
EE 531 and EE 532 (4) Engineering Electronics
or
Phy 361 (4) Modern Electronics
and
Phy 562 (3) Modern Electronics
and
Phy 563 (3) Modern Electronics
EE 535 (4) Digital Instrumentation
and
EE 536 (3) Digital Instrumentation
ME 553 (3) Engineering Hemodynamics
Phil 522 (5) Symbolic Logic
or
Math 304 (4) Mathematical Logic

CS courses at the 600 level upon consent of the instructor and the student's undergraduate advisor.

*A student who elects CS 428 and CS 529 may not apply CS 521 and CS 522 toward the degree requirements.

****

Students planning their programs should elect certain mathematics courses prior to the core courses in Computer Science. In particular, Math 112 is a prerequisite to CS 105 and Math/CS 334 is a prerequisite to CS 321. The

calculus sequence (Math 111, 112, 113, 321) should be completed without interruption once begun.

Students who have taken CS 101 and Math 113 and who delay their decision to become Computer Science majors until the beginning of their junior year (such as taking the common two-year engineering program) may possibly complete the CS requirements in two years of further study (see Program B below). As an alternative to the CS 105 requirements, a student may complete three hours of CS approved electives with advisor's permission. However, CS majors are strongly urged to take CS 105 rather than CS 101 if at all possible. CS 101 is designed primarily for the non-major who wishes to learn more about computer science outside his major. Late majors who have not taken CS/Math 334 should take it in the summer before CS 321.

3.5 PRECEDENCE DIAGRAM FOR REQUIRED CS COURSES

Autumn	Winter	Spring
	334	105
321	322	323
	427	
306	411	412
	417	

3.6 COMPUTER SCIENCE AS A SECOND DEGREE

Some students may wish to earn a degree in Computer Science as their second B.S. degree. The Department of Computer Science requires only that the requirements of this Department are met, the minor requirement being fulfilled by the student's other major. In some cases fewer than 51 additional hours are required because of overlaps in the requirements of the first and second degree. This is especially true of students in Computer Science and Electrical Engineering. The Department of Computer Science should be notified of a planned double major.

3.7 UNDERGRADUATE FINANCIAL ASSISTANCE

The Department of Computer Science of the University of Utah has the following scholarship and financial assistance awards available:

SPERRY RAND SCHOLARSHIP - This scholarship is awarded on the basis of excellence to one Junior and one Senior student majoring in Computer Science. The award is an unrestricted $500 cash prize and is unrelated to any other financial support the selected student may be receiving. Two awards are made

each year.

SPECIAL DEPARTMENTAL SCHOLARSHIP - This award is available to entering Freshmen or transfer students who plan to major in Computer Science and are residents of the state of Utah. The award is for resident tuition for three quarters, renewable up to twelve quarters, depending on GPA, for undergraduate work. There is one award available each year.

UNDERGRADUATE TEACHING ASSISTANTSHIP - The Department employs a number of Junior and Senior students as undergraduate teaching assistants. This assistantship involves 20 hours of work per week (usually grading and consulting) at an appropriate hourly wage.

CONTINUING STUDENT DEPARTMENTAL SCHOLARSHIP - This award is available to Sophomore, Junior or Senior students majoring in Computer Science and who are residents of the state of Utah. The award is for resident tuition for three quarters, renewable up to six quarters, depending on GPA, for undergraduate work. There are three awards available each year.

The Department also collects and forwards applications for the following College of Engineering Awards:

JOSEPHINE BEAM EDUCATIONAL SCHOLARSHIP - This award is available to engineering students in the College of Engineering and the College of Mines. The award is for approximately $500, with twenty-seven being awarded each year. The decision on this award is based on financial need. Selection is made by the College of Engineering and the Financial Aids Office.

CLYDE L. CHRISTENSEN SCHOLARSHIP - This award is available to College of Engineering students. The award is for approximately $400, with twenty awards given each year. Selection is made by the College of Engineering.

The deadline for application for 1981 is February 1. For applications write to:

Scholarship Committee
c/o Loretta J. Cruse
Department of Computer Science
3160 Merrill Engineering Building
University of Utah
Salt Lake City, Utah 84112

For additional information on financial aid and other scholarships contact:

University of Utah
Financial Aids Office
Salt Lake City, Utah 84112

3.8 EMPLOYMENT OPPORTUNITIES

A registry of Computer Science students who wish part-time employment while they are in school is kept by a professor in the Department of Computer Science. Employers who request names of qualified students from the Department will be given the names and telephone numbers of students who wish employment in that area of work.

Students seeking employment upon graduation should contact the University Placement Center in order to be included on a list supplied to employers. Students not planning to work towards an advanced degree should register with the Placement Center in their Junior year, since most companies begin interviewing in the Autumn Quarter of the Senior year.

STUDENT PARTICIPATION IN DEPARTMENTAL AFFAIRS

Opportunities for students to develop their organizational and leadership abilities are available through participation on the Student Advisory Committee (SAC) which coordinates the following:

1. Course and teacher evaluations.
2. Representation (3 students) on the Joint Engineering Council (makes decisions affecting students in the College of Engineering).
3. Representation (1 student) on the Campus SAC Council.
4. Engineering Week in February.

Student input is continually being sought regarding items affecting students; e.g., scheduling and class conflicts, two hour block courses, new courses, etc. A mailbox in 3162 MEB is provided for initial contact with SAC. Any and all criticisms, suggestions, or participation is solicited.

4. INFORMATION FOR GRADUATE STUDENTS

4.1 DEGREE REQUIREMENTS

Ph.D. DEGREE:

The Ph.D. is a research degree offered through the Graduate School. It is awarded to a candidate who has demonstrated breadth in Computer Science in general, and depth in a research specialty within Computer Science. Most of a Ph.D. student's time is devoted to courses and research, including personal participation in the research and teaching environment of the Department on a day-to-day basis. Candidates must complete 54 credit hours of graduate coursework and 20 hours of dissertation research. The time for completion of the Ph.D. is normally four years of graduate study, assuming some teaching obligations during that time.

M.S. DEGREE:

The M.S. is a research degree offered through the Graduate School. Candidates must devote a minimum of 45 quarter hours to graduate courses and thesis. At least 36 quarter hours must be in resident study at the University of Utah. A minimum of 30 quarter hours must be in coursework with the balance in thesis hours. A full-time student working on an M.S. program normally completes the degree requirements, including thesis, within two calendar years.

M.E. DEGREE

The M.E. is a professional, non-thesis Master's degree offered by the College of Engineering through the Department of Computer Science. This degree requires at least 45 credit hours of coursework, including a minimum of 27 hours of 500 level and above Computer Science courses.

M.PHIL. DEGREE

This degree requires the same qualifications for admission and scholarly achievement as the Ph.D. except that it does not require a doctoral dissertation. All regulations covering the Ph.D. degree apply to the M.Phil. degree. This degree, like the Ph.D. degree is a terminal degree; a student will not be considered a candidate for both degrees in the same department.

4.2 INSTRUCTIONS FOR APPLICATION TO GRADUATE STUDIES

I. The following material is required from all applicants:

a. Fill out and return the University of Utah "Application for Admission to Graduate School" form, along with the application fee to the University Admissions Office, not to the Department of Computer Science.

b. In accordance with the instructions on the Application form, arrange to have two copies of your transcripts sent to the University Admissions Office directly from the issuing schools.

c. Arrange to have three letters of recommendation sent directly to the Department of Computer Science Graduate Admissions Chairman (in care of Mrs. Ramona Rei). We prefer recommendations from former teachers.

d. Send a letter to the Department of Computer Science Graduate Admissions Chairman (in care of Mrs. Ramona Rei) describing in depth your background, interests, and particularly your reasons for wanting to pursue graduate studies in Computer Science at the University of Utah.

e. Take the GRE (aptitude and advanced in either Computer Science or Mathematics) and have the scores reported to the Department of Computer Science.

f. Applicants whose native language is not English should take the "Test of English as a Foreign Language" (TOEFL) Examination and have the score reported to the University of Utah Admissions Office.

For admission in the Autumn Quarter, August 1 is the deadline for receipt of application materials in the University of Utah Admissions Office. Please allow for about two month's delay in processing your application after receipt in the Admissions Office. Foreign applicants should allow for four month's delay.

II. Special information for foreign applicants:

The University Admissions Office frequently delays applications from foreign applicants they believe to be incomplete. For your own protection, please notify us as soon as you feel your application is complete, so that we may contact the Admissions Office for your file. Letters of recommendation and your own personal letter must be written in English.

ADDRESSES

Chairman of Graduate Admissions
c/o Loretta J. Cruse
Department of Computer Science
University of Utah
Salt Lake City, Utah 84112

Admissions Office
University of Utah
309 Park Building
Salt Lake City, Utah 84112

4.3 GRADUATE STUDENT FINANCIAL ASSISTANCE

I. Assistantships in Computer Science

There are three types of financial aid available to graduate students in the Department of Computer Science: 1) teaching assistantships, 2) departmental research assistantships, and 3) research assistantships available from individual faculty research investigators. Other forms of financial aid such as National Foundation Fellowships, should be requested directly from the granting agency. Below are the definitions of these positions:

a. Teaching Assistantship

A teaching assistant is a regularly enrolled student employed .25 to .50 fte (full-time equivalent) for assisting a faculty member in teaching. The teaching assistant is required to meet with students regularly in a classroom, laboratory, or other instructional setting; to prepare for instructional duties through lesson and materials preparation; to counsel students outside of the regularly scheduled instructional periods; and to evaluate and grade students' work to aid in the determination of the students' total course grade. Teaching assistants are eligible for such benefits as a reduced rate at the University Bookstore, extended check-out privileges at the University Libraries, and an option to purchase an "A" parking sticker.

b. Research Assistantship

The Graduate Committee assigns each research assistant to a particular faculty member based on mutual interest and agreement as expressed by the student on his or her application for financial aid and verbally by the professor reviewing the applications. The duties assigned to a research assistant should be in line with the student's research interests and also useful to the professor's research efforts. A research assistantship can be viewed as a kind of internship learning by practicing under faculty

supervision. A student wishing to be research assistant should contact the faculty member directly and make arrangements on an individual basis.

Summer Support

Faculty members who are supervising research projects often hire students for half or full-time during the summer. Students interested in a summer appointment as a research assistant should make arrangements with the faculty member directly. There are few teaching assistantships available during the summer.

II. Computer Science Graduate Student Pay Scales

It is the goal of the Computer Science Department to be able to offer up to seven quarters of departmental financial support on the Ph.D. level and up to four quarters on the terminal Master's level to students who are progressing satisfactorily toward completion of their degree. Funding will, of course, be contingent on yearly departmental funding. The department will notify students of funding awards as soon as possible before the start of a quarter. It is usually not possible, however, to notify students of their specific assignments (that is, whether they will be an RA or TA) until the beginning of each quarter, as these decisions are based on class enrollment figures. Listed below is the pay scale for Computer Science graduate students who were RAs or TAs during 1979-80 based on .5 fte. (Undergraduate TAs are usually paid on a lower scale than graduate students.)

Level 1. Graduate students entering the program with a Bachelor's degree---------- $475/mo.

Level 2. M.S. and Ph.D. students who have successfully passed their respective comprehensive examination---------- $550/mo.

Level 3. Ph.D. students who have successfully completed the oral qualifying exam and have their approved thesis proposal on file---------- $625/mo.

Entering students who possess a Master's degree will start at Level 2, $550/mo. It is anticipated that these rates will be revised periodically as needed. Below are some explanatory notes on the pay policy:

a. Pay level changes are made between quarters. It is the student's responsibility to notify the main office of his

eligibility for a pay raise.

b. In order for a student to be raised to Level 3 support, a copy of his or her Ph.D. research proposal, signed by all members of a student's supervisory committee, must be on file in the main office.

c. Full-time students who are employed by the University are exempt from FICA taxes and should file a form to this effect.

III. Tuition Waivers

Only Computer Science graduate students who are appointed as TAs are eligible for tuition waivers. Waivers are awarded on a quarter-by-quarter basis. Notification of awards will normally be made as soon as possible before the start of the quarter. Information pertinent to this judgment includes items such as: performance in courses, past performance as a TA or RA, or, for new students, previous academic records and letters of recommendation.

IV. Procedures for Applying for Financial Assistance

Qualified full-time graduate students are eligible for financial aid in the form of teaching assistantships or research assistantships as described above. A general purpose financial assistance form can be obtained in the Computer Science Department as well as from the Graduate Fellowship Office, 312 Park Building. This completed application should be submitted directly to the Computer Science Department by March 1 for aid to commence the following Autumn quarter.

COMPUTER SCIENCE COURSES

101. PROGRAMMING WITH FORTRAN (3) A W S Prereq: Math 105. An introduction for non-majors to programming using the high-level language FORTRAN. Emphasis on laboratory practice and structured problem-solving techniques.

102. PROGRAMMING WITH COBOL (3) A Prereq: Math 105. An introduction for non-majors to programming using the high-level language COBOL. Emphasis on laboratory practice and structured problem-solving techniques.

103. PROGRAMMING WITH BASIC (3) A W S Prereq: Math 105. An introduction for non-majors to programming using the high-level language BASIC. Emphasis on laboratory practice and structured problem-solving techniques.

105. INTRODUCTION TO COMPUTER SCIENCE (3) W S Prereq: Math 112. The first course for students majoring in Computer Science. Algorithms, their flowchart representation, construction and analysis. Semantics of algorithms based on abstract and concrete computer models. Stepwise decomposition and structured programming for composing correct algorithms. Data structures with emphasis on trees and linked lists. Data structure accessing methods. String processing. Laboratory practice in mapping algorithms to computer programs in languages having various levels and types of primitives.

301. THE ART OF INFORMATION PROCESSING (3) A S Prereq: Sophomore standing. The history and the fundamental concepts of information processing. The social, political and philosophical impact of the information revolution. Discussion of dehumanization, privacy, social control. The computer in literature and non-technical applications. Intended primarily for students in the arts, humanities, and social and behavioral sciences. (Not offered 1980-81).

306. INTRODUCTION TO PROGRAMMING LINGUISTICS (3) A Prereq: CS 105. Semantics and syntax of programming languages. Principles of interprocedure communication and recursion. Use of a descriptive language and an interpretive model as a base for studying the comparative semantics of a variety of languages such as ALGOL, FORTRAN, LISP, and SNOBOL. Snapshots and other tracing techniques, manual and automatic. Introduction to multitasking. Laboratory practice programming in several of the above languages.

321,322,323. FOUNDATIONS OF DIGITAL COMPUTER ORGANIZATION (4,4,4) A W S Prereq: CS 105, Phys 171, CS/Math 334. Fundamental electronic devices, network theory, transistor models and operation, gates and switching elements, switching algebra, combinational and sequential circuit synthesis and analysis, memory systems, I/O systems, peripheral device organization, processor subsystems, computer arithmetic, coding, computer architecture,

microprocessors, microprogramming, and machine level programming.

334. DISCRETE STRUCTURES (also Math 334) Su W Prereq: Math 105. An introduction to formal logical arguments. Finite sets, relations, functions, graphs, semigroups, groups, applications to Computer Science. Elements of elementary logic.

335. DISCRETE STRUCTURES (also Math 335) (3) S Prereq: CS/Math 334. Lattices, Boolean algebras, special topics (e.g., combinatorics, algorithms), applications to Computer Science. More elements of logic.

376. INTERACTIVE NUMERICAL METHODS (3) S Prereq: Math 351. An introduction to interactive numerical problem-solving for science and engineering students. Lectures and problems will emphasize basic numerical techniques and their application to scientific problems. Topics include: computer fundamentals, algorithms, flowcharts and programming in BASIC, finite differences, interpolation, integration and smoothing, curve fitting and orthogonal functions, solutions of ordinary and partial differential equations, solution of linear equations, solutions of non-linear equations (root finding), simple statistics and Monte-Carlo methods.

395. SPECIAL STUDIES FOR UNDERGRADUATES. (Arr.) Prereq: consent of instructor. Topics will be announced.

410. LOW LEVEL COMPUTER PROGRAMMING (4) A Prereq: CS 323 and CS 417 Introduction to lower level languages (assembler languages, machine codes, microcode) with emphasis on macroprocessors.

411,412. PROGRAMMING LINGUISTICS AND DATA STRUCTURES (4,4) W S Prereq: CS 306, CS/Math 334. Theory and advanced techniques in programming linguistics and data structures. Multitasking, including interprocess communication and synchronization. High-level data structure definition and manipulation. Introduction to language extensibility. Analysis of the theory and implementation of a pedagogic language. Syntax description via grammars and abstract machines; parsing algorithms. Introduction to processing large data structures, including models, access languages and their implementation.

417,418. PROGRAMMING LABORATORY (4,4) W S Prereq: CS 306 for 417, CS 411 and 417 for 418. Practical experience in the design, development, documentation, and implementation of significant software modules using PASCAL and microcode.

427,428. DIGITAL SYSTEMS LABORATORY (4,4) W S Prereq: CS 321 for 427; CS 322 and 427 for 428. Also recommended for students in CS 411. Experiments

with electrical and logical behavior of switching devices. Design projects.

490. SENIOR PROJECT (1-3) W S Prereq: Senior standing and consent of instructor. This course must be taken in two consecutive quarters with 1 credit hour for the first quarter and 3 credit hours for the second quarter. Attendance is required for both quarters. Students select project topics and submit proposals under consultation with the instructor in the first quarter. The proposed projects are implemented and oral and written engineering reports are given during the second quarter.

495. SPECIAL STUDIES FOR UNDERGRADUATES (Arr.) Prereq: Consent of instructor. Topics will be announced.

501. A SURVEY OF PROGRAMMING THEORY AND PRACTICE (4) A Prereq: CS 101 or equivalent. For persons having some technical training, modern software design and for implementation techniques. May be taken for graduate credit by non-majors, and taken without credit as an intensive review for CS majors. Topics review for CS majors. Topics covered include: Algorithm and analysis; stepwise decomposition and structured programming; data structures with an emphasis on trees and linked lists; recursive programming; tree search algorithms; string processing; syntax and semantics of programming languages, principles of interprocedure communication; use of abstract machine interpreters for program analysis tracing and debugging techniques; conventional computer organizations; support for programming (machine arithmetic, memory systems, registers, stacks, addressing mechanisms etc.), laboratory experience.

502. A SURVEY OF MODERN COMPUTER LANGUAGES (4) W Prereq: CS 501. Builds on CS 501 and is for non-majors who wish additional experience with modern programming techniques and languages. (Topics listed for CS 501 but not covered may be covered in this course). The course will concentrate on the special problem areas addressed by at least three of the following languages: ALGOL, COBOL, FORTRAN, LISP, PASCAL, and SNOBOL. Other languages may be included as appropriate. Emphasis will be placed on the comparison features, natural problem domains, laboratory experience, and problem solving techniques.

511, 512. PROGRAMMING LINGUISTICS AND DATA STRUCTURES (4,4) (for graduate students) W S Prereq: CS 306, CS/Math 334. Theory and advanced techniques in programming linguistics and data structures. Multitasking, including interprocess communication and synchronization. High-level data structure definition and manipulation. Introduction to language extensibility. Analysis of the theory and implementation of a pedagogic language. Syntax description via grammars and abstract machines; parsing algorithms. Introduction to processing large data structures, including models, access languages and their implementation. A student may not get credit for both 411 and 511, nor both 412 and 512.

519. PROGRAMMING LABORATORY (4) A Prereq: CS 412 and 418. Practical experience in the design, development, documentation, and implementation of significant software modules. Individual or team projects.

520. COMPILER CONSTRUCTION (3) W Prereq: CS 410 or 417, and 412/512 or consent of instructor. Engineering principles of compiler and subroutine packages. The course builds on theory presented in CS 412/512 (or equivalent): top-down and bottom-up parsing, recursive descent; LL(k) and LR(k) parsing; precedence; lexical analyzers; symbol tables; internal forms and intermediate languages; code generation; code optimization, semantic specifications; error detection and recovery; comparison of methods. A number of compilers and interpreters for simple languages will be written.

521. PROGRAMMER'S INTRODUCTION TO LOGIC DESIGN (also EE 576) (3) S Prereq: CS 306 or CS 502. An introduction to the hardware logic design of digital systems and interfaces, using a top-down methodology, based on programming principles. Stated problems, whose solutions are expressed as algorithms, are mapped with the aid of an appropriate hardware description language into state machines. Implementation of these state machines will use selected MSI and LSI microelectronic building blocks. CS 522 is the laboratory part of this course and should be elected concurrently or immediately following this course. A student who elects CS 428 and CS 529 may not apply CS 521 and CS 522 toward the degree requirements.

522. LOGIC DESIGN LABORATORY (2) S Prereq: CS 521 or concurrent registration in CS 521. Laboratory practice in the hardware realization of algorithms using disciplines introduced in CS 521. This laboratory should be taken concurrently with CS 521 or in the following offering. A student who elects CS 428 and CS 529 may not apply CS 521 and CS 522 toward the degree requirements.

523. PROGRAM VERIFICATION (3) W Prereq: CS 534 or CS 334 and 411/511. The concepts underlying verification, the relationship between choice of semantic structure and ease in verification, and the relative power of current verification methods. Models: Algol and flowchart programs, transition systems, non-deterministic programs, grammars, parallel systems of recursive functionals, program graphs, abstract syntax. Correctness properties: invariance, partial correctness, termination, deadlock-freedom, boundedness, equivalence. Proof methods: transition induction, structural induction, sub-goal induction, reduction, predicate transformation, least fixed point induction. (Not offered 1980-81)

524. SWITCHING CIRCUIT THEORY (4) W Prereq: CS 321 or consent of instructor. Topics in combinational circuits. Finite state machines; state and machine identification experiments; fault detection experiments. Memory aspects of machines: definiteness, finite memory, information losslessness. Decomposition of machines. Linear machines. Applications to codes: uniquely

decipherable codes, error correcting codes.

529. DIGITAL PROJECT LABORATORY (4) A Prereq: CS 323 and 428. A project course for advanced digital design students.

534. INTRODUCTION TO THEORETICAL COMPUTER SCIENCE (3) A Prereq: Graduate standing. Survey of discrete mathematical concepts relevant to Computer Science. Sets, relations, functions, propositional calculus, predicate calculus, graphs, partial orders, equivalence relations, enumeration, inductive definitions and proofs, semigroups, groups, and lattices as applied to state-transition systems, finite-state machines, regular expressions, languages, grammars, Turing machines, partial recursive functions, and other simple computation models. Emphasis will be placed on the type of reasoning necessary to construct sound mathematical proofs.

536. OPERATING SYSTEMS (4) S Prereq: CS 323 and 511. Characteristics, objectives and issues concerning computer operating systems. Process implementation, synchronization, memory management, name management, protection, resource allocation, system modeling, pragmatic aspects, case studies.

539. FUNDAMENTALS OF INTEGRATED CIRCUITS (also EE 560) (4) A Prereq: CS 427, EE 221. Overview of currently important LSI processes such as MOS, bipolar, I^2L and TTL. N channel-silicon gate-depletion load process, and the art and science of integrated circuit design using this process. Designs of SSI and MSI level circuits and some simpler LSI circuits. Exercises will provide experience in creating topological and composite layouts for circuits at the LSI level. Selected circuits will be designed and fabricated.

540. MODELLING OF INTEGRATED CIRCUITS (also EE 561) (4) W Prereq: CS 539, EE 560 or consent of instructor. Computer modelling of integrated circuits including LSI circuits. Modelling of timing problems, verification of correctness, performance enhancement.

541. SIMULATION OF PHYSICAL SYSTEMS (also EE 572) (3) A Prereq: A computer programming course or equivalent experience; Phys 173, Math 351. A problem solving course in which the technique of computer simulation is learned and applied to the solution of engineering problems. Analysis and synthesis associated with continuously varying systems are considered. The behavior of such systems can be described by one or more ordinary or partial differential equations having linear, non-linear and time varying aspects. Analog computers and digital computers using special simulation languages are utilized.

542. SIMULATION OF DISCRETE STATE SYSTEMS (also EE 577) (3) W Prereq: CS 101 and elementary probability theory or consent of instructor. Underlying principles of discrete system simulation and the programming of digital computers for the simulation of these systems. Both general purpose programming languages and special purpose simulation programming languages will be employed. Depending on the needs and interests of the class, discrete systems will be simulated using general purpose languages such as FORTRAN, ALGOL, or PPL, and special purpose languages such as GPSS, GASP or simscript.

543. ADVANCED DISCRETE STATE SYSTEM SIMULATION (3) S Prereq: CS 542. A continuation of CS 542. Emphasis on the practical aspects of discrete system simulation. Systems of greater complexity and sophistication will be studied. Techniques for verification of simulation results.

551. COMPUTER GRAPHICS (3) A Prereq: Math 560 or 561. Basic display techniques, display devices, vector generation, display processors. Homogeneous coordinates, transformations and clipping in 2D and 3D. Graphics systems, interactive graphics. Introduction to raster graphics. Some elements of photography as related to computer graphics.

561. INTRODUCTORY WAVEFORM PROCESSING (Also EE 517) (4) A Prereq: Ordinary differential equations or electric circuit theory or linear algebra or consent of instructor. Studies basic to the digital essing of analog signals by means of computers. Fourier transform theory. Convolution and superposition. The relation of time and frequency response through Fourier transforms. Sampling theorem. Development of the discrete Fourier transform, and its relation to the Fourier transform. Laboratory instruction provided.

562. ADVANCED WAVEFORM PROCESSING (Also EE 518) (4) W Prereq: CS 561. Discrete systems. Z transform theory. Relation between the Z transform and the discrete Fourier transform. The Fast Fourier Transform. Fast algorithms for Fourier analysis, convolution, correction and spectral estimation. Hilbert transform theory. Laboratory instruction provided.

563. SENSORY INFORMATION PROCESSING (Also EE 519) (4) S Prereq: CS 562 or consent of instructor. Analysis of two dimensional signals. Design of finite impulse response and infinite impulse response digital filters. Nonlinear processing of sound and images using homomorphic signal analysis. Speech analysis and data compression techniques. Laboratory instruction provided.

565. COMPUTER MUSIC SEMINAR (2) W Prereq: CS 101 or 105. Basic concepts of musical instruments, performance and composition in relation to corresponding electrical signals and coding. Use of trigonometric, exponential and logarithmic functions as a basis for generating sound. Students will have an opportunity to generate sound and musical samples on the computer.

566. COMPUTING WITH SYMBOLIC EXPRESSIONS (3) W Prereq: CS 412/512 or consent of instructor. An introduction to list and string processing with applications to symbolic differentiation, simplification of algebraic expressions, and translators and compilers. Design of list processing systems.

570-579. TOPICS IN COMPUTER SCIENCE (Arr.) Prereq: Consent of instructor. May be repeated for credit.

590. INDEPENDENT STUDY (Arr.) Prereq: Consent of instructor and the student's undergraduate advisor. Special reading and/or projects. May be repeated for credit.

591-599. SEMINAR (Arr.) Prereq: Consent of instructor. Current topics in Computer Science. Topics will be announced. May be repeated for credit.

610. FORMAL LANGUAGES (4) A Prereq: CS 512, CS 534 and Math/CS 334. Properties of various classes of formal languages and the grammars which generate them. Relevance to the syntactic structure and parsing of programming languages will be discussed. (Not offered 1980-81).

612. COMPUTATIONAL COMPLEXITY (3) S Prereq: CS 534. Analysis of time and memory requirements of algorithms for sorting, set manipulation, graph analysis, matrix operations, arithmetic, Fourier transforms, and pattern matching. NP complete problems. Complexity hierarchies.

628. ORGANIZATION OF COMPUTING SYSTEMS (4) W Prereq: CS 323 or CS 521. Existing and proposed computers are studied with a view towards gaining a comparative understanding of the merits and shortcomings of various types of computer architectures.

631,632. SOFTWARE ENGINEERING (4,4) W S Prereq: CS 410 and 519 or consent of instructor. Study and application of design methodology, support tools and implementation pragmatics of large programming systems. Aspects of modularity, specification, portability, evaluation and other topics of current interest. Individual and group projects leading to high quality software.

638. DATA BASE SYSTEMS (4) A Prereq: CS 512. Modeling of real world structures and their mapping into relational, network, and hierarchic schemata; the design and implementation of data base systems including integrity, security and concurrency control; programming experience on a commercial data base system using both data definition and data manipulation languages. Coverage ranges from proven practical techniques to current research activities.

641. LARGE SCALE INTEGRATED CIRCUITS DESIGN (also EE 662) (3) S Prereq: CS 428 and CS 540. Design of microprocessors, RAM, ALU, PLA, and SLA (Storage/Logic Arrays). An LSI design completed by students will be fabricated and tested.

642. DESIGN OF LARGE DIGITAL SYSTEMS (3) S Prereq: CS 529. Problems encountered in large system design, and strategies and proven methodologies used in the design of such systems. Lectures will include specialists from industry. Illustration of designs of specific large digital systems.

647. ARTIFICIAL INTELLIGENCE (3) W Prereq: CS 334; CS 512 recommended, but not required. Attempts to produce computers exhibiting intelligent behavior will be studied. Topics will be selected from the following: heuristic programming, problem solving, theorem proving, question answering, machine learning, pattern recognition, game playing, philosophical and social issues.

652. COMPUTER GRAPHICS (3) W Prereq: CS 551. Representations of three-dimensional objects, polygons, three-dimensional visualization techniques, hidden line and hidden surface removal, polygon clipping, continuous tone pictures, color displays, lighting models, the aliasing problem. Some fundamentals of photographing computer generated color images.

653. COMPUTER GRAPHICS (3) S Prereq: CS 652. A project course offering more detailed treatment of topics selected from those covered in CS 652. Study of most recent developments in computer graphics. Further photographic studies including computer generated motion pictures.

654. COMPUTATIONAL PHYSICS (3) A Prereq: Undergraduate physics degree or graduate standing in Computer Science. Advanced numerical programming techniques. General methods for solving partial differential equations, applications to problems in gravitation, electricity and magnetism, plasma cations. Automatic programming of numerical problems. (Not offered 1980-81).

656. COMPUTATIONAL PHYSICS (3) S Prereq: Undergraduate physics degree or graduate standing in Computer Science. Problem solving by symbolic computation. Polynomial manipulation, rational functional manipulation, pattern-matching techniques, linear equations, analytic integration. Solution of numerical problems by symbolic methods. (Not offered 1980-81).

667,668,669. COMPUTER AIDED GEOMETRIC DESIGN (also Math 667, 668, 669) (3,3,3) A W S Prereq: Math 560 or 561. Coreq: CS 551. Representation and approximation of curves and surfaces for computer aided modeling; splines and their variational properties, tensor product interpolants, interpolants over triangles, Coons patches; interactive design schemes.

670-679. ADVANCED TOPICS IN COMPUTER SCIENCE (Arr.) May be repeated for credit.

680-689. SEMINAR (Arr.) Prereq: Consent of instructor. Current topics in Computer Science. Topics will be announced. May be repeated for credit.

Special seminars offered in prior years are:

1. Functional Programming Languages
2. Software Portability and Small Machines
3. Abstract Machines
4. Computer Animation
5. Computer Vision and Manipulation
6. Petri Net Theory
7. Semantics of Concurrent Systems
8. Reduction Machines

690. INDEPENDENT STUDY (Arr.) Prereq: Consent of instructor. May be repeated for credit.

697. THESIS RESEARCH (Master's)(Arr.).

698. RESEARCH CONSULTATION (Master's)(3).

699. CONTINUING REGISTRATION (Master's)(0).

780-789. SEMINAR (Arr.) May be repeated for credit.

797. THESIS RESEARCH (Ph.D.) (Arr.).

798. RESEARCH CONSULTATION (Ph.D.)(3).

799. CONTINUING REGISTRATION (Ph.D.)(0).

5. CS FACULTY AND THEIR RESEARCH INTERESTS

ROBERT E. BARNHILL, Professor of Computer Science and Professor of Mathematics; Ph.D., University of Wisconsin, 1964

Professor Barnhill joined the University of Utah Mathematics faculty in 1964. He was appointed to his current rank as Professor of Computer Science in 1976. He is interested in computer aided geometric design; the approximation and representation of surfaces; the finite element analysis of elliptic boundary value problems; and interpolation and integration of functions of more than one variable. He and his co-workers have developed several new methods for interpolation and integration of functions of more than one variable and for interpolation over triangulated regions. He has also recently developed computable error bounds for finite element analysis. He has been a visiting professor at Brown University, General Motors Research Laboratories, George Washington University, and Brunel University, England. His professional activities include Chairman and organizer of Office of Naval Research Navy Workshop in Numerical Analysis on "The Approximation of Functions of More than One Variable," Annapolis, 1971; International Coordinator for SIGNUM (Special Interest Group in Numerical Mathematics); invited speaker in the United States of America and abroad, and referee and reviewer for the relevant agencies and journals. He has directed Ph.D. and Master's theses on a wide range of numerical analysis topics, the most recent ones being in computer aided geometric design. He initiated the "Calculus with Computing" course in the Department of Mathematics in 1965.

Representative Publications:

Barnhill, R.E., and Gregory, J.A., 1975. Polynomial Interpolation to Boundary Data on Triangles. Mathematics of Computation 19, pp. 726-735.

Barnhill, R.E., 1975. Smooth Interpolation Over Triangles. Computer Aided Geometric Design, pp. 45-70. Academic Press, New York.

Barnhill, R.E., and Brown, J.H., 1976. Nonconforming Finite Elements for Curved Regions. Proceedings of the Numerical Analysis Conference at Dundee University, G.A. Watson (ed.), pp. 1-14, Springer-Verlag Lecture Notes on Mathematics, no. 506.

BRENT S. BAXTER, Research Assistant Professor of Computer Science and Radiology; Ph.D., University of Utah,

Professor Baxter's research activities are concerned with characterizing and modeling phenomena important to human vision, such as neural inhibition in the retina, masking by narrow band noise, brightness constancy, geometric

illusions and depth perception. An understanding of these phenomena can often be used to advantage in devising improved image manipulation systems, especially in diagnostic medicine where the input may be a poorly visualized anatomical or pathological structure. Dr. Baxter has constructed a 3-dimensional display for viewing x-ray computed tomograms of the head and abdomen, and equipment for producing high fidelity transparency films from digital scintillation camera images. As part of these activities he has developed a special purpose programming language for manipulating digital image data.

Representative Publications:

Baxter, B.S., 1975. Image Processing in Human Vision, DARPA Technical Report UTEC-CS-75-168.

Frei, W., and Baxter, B.S., 1977. Rate-Distortion Coding Simulation for Color Images. IEEE Trans on Com. Vor. COM-25 11 1977.

Baxter, B.S., Precision Computer Display Techniques in Nuclear Medicine. Proc. 23 International Symposium. SPIE. San Diego, 1979.

Koehler, P.R., Anderson, R.E., Baxter, B.S., The Effect of Computed Tomography Viewer Controls on Anatomical Measurements, Radiology, Vol. 1 #1, pp. 189-194, Jan. 1979.

STEVEN F. BOLL, Assistant Professor of Computer Science; Ph.D., University of Utah, 1973.

Professor Boll is engaged in research in digital signal processing. Specifically he is investigating efficient filter design methods for acoustic noise suppression; robust real time, digital speech bandwidth compression algorithms; and spectral analysis algorithms.

Representative Publications:

Boll, S.F., "Adaptive Noise Cancellation in Speech Using the Short Time Fourier Transform, proc. of the 1980 International Conference on Acoust. Speech and Signal Processing, Denver, Colorado, April 1980.

Boll, S.F., "Suppression of Acoustic Noise in Speech Using Spectral Subtraction," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-28, April 1979.

Youngberg, Jim and Boll, S.F., "Constant Q Signal Analysis and Synthesis." Proc. of the 1978 International Conference on Acoustics Speech and Signal Processing, Tulsa, Oklahoma, April 1978.

ELAINE COHEN, Research Assistant Professor of Computer Science and Adjunct Assistant Professor of Mathematics; Ph.D., Syracuse University, 1974.

Professor Cohen did her undergraduate work in Mathematics and Physics at Vassar College. Her M.A. and Ph.D. are from Syracuse University in Mathematics. Her doctoral thesis is in the area of Real and Functional Analysis, an area of continuing research for her. Since joining the Department of Computer Science, she has been exploring applications of mathematical structures to problems in sensory information processing, part of a research effort to develop a more unified explanation of sensory phenomena. Other interests include computer aided geometric design, coding theory, and aspects of numerical analysis.

Representative Publications:

Cohen, E., On the Degree of Approximation of a Function by the Partial Sums of its Fourier Series. Transactions of the American Mathematical Society, Vol. 235, Jan. 1978, pp. 35-74.

Cohen, E., On the Fourier Coefficients and Continuity of Functions of Class.

Cohen, E., and Riesenfeld, R.F., An Incompatibility Projector Based on an Interpolant of Gregory.

ALAN L. DAVIS, Associate Professor of Computer Science; Ph.D., 1972, University of Utah.

Dr. Davis' current research interests include distributed computer architecture, graphically specified concurrent programming languages, parallel program schemata, device integration, asynchronous circuits, and self-timed systems. His laboratory is a hardware prototyping facility which also contains DDM2, a fully asynchronous, prototype, data-driven computer. DDM2 is currently connected by a custom link to the DEC-20. The Data-Driven Research Project is conducting both architectural and operating systems experiments on DDM2 to determine its usefulness as a general purpose processing element in a distributed control environment. The laboratory also contains 3 graphics systems which support graphical programming studies. Dr. Davis was a National Academy of Sciences visitor to the Soviet Union, where he spent three months at the Siberian Academy of Sciences Novosibirsk Computing Center.

Representative Publications:

Barton, R.S., Davis, A.L., et al., System and Method for Concurrent and Pipeline Processing Employing a Data-Driven Network. U.S. Patent 3,978,452, issued Aug. 31, 1976.

Davis, A.L., Principles for Distributed Control Computer Architecture, Proceedings of the Rocky Mountain Symposium on Microcomputers, 1978.

Davis, A.L., The Architecture and System Method of DDM1: A Recursively Structured Data-Driven Machine. Proceedings of the Fifth Annual Symposium on Computer Architecture, 1978.

Davis, A.L. DDN's - A Low-Level Program Schema for Fully Distributed Systems. Proceedings of the First European Conference on Parallel and Distributed Processing; pp. 1-7; Toulouse, France (Feb. 1979).

Davis, A.L. A Data-Driven Machine Architecture Suitable for VLSI Implementation. Proceedings of the Caltech conference on Very Large Scale Integration: Architecture, Design, Fabrication. pp. 479-494 (January 1979).

Davis, A.L. A Data-Flow Evaluation System Based on the Concept of Recursive Locality. Proceedings NCC '79. pp. 1079-1086, New York (June 1979).

DAVID C. EVANS, Adjunct Professor of Computer Science (Resident at Evans and Sutherland Computer Corporation); Ph.D., University of Utah, 1953.

Professor Evans is presently engaged in research in computer models and image synthesis. He is also co-founder and President of the Evans and Sutherland Computer Corporation. He was formerly Professor of Electrical Engineering at the University of California, Berkeley, and Director of Engineering of the Computer Division of the Bendix Corporation. During his career he has been group leader in the development of the G-15, G-20, and the SDS-940 computing systems. Professor Evans joined the Computer Science faculty in 1966.

ERCOLINO FERRETTI, Research Associate Professor of Computer Science; B.M., New England Conservatory of Music, 1950.

Professor Ferretti is presently conducting research in computer music. The research is closely tied to the course CS 565, which is interdisciplinary in nature, with an emphasis on artistic expression coupled to modern technology.

Professor Ferretti graduated from the New England Conservatory of Music with a major in Composition and was awarded the President's Prize for Composition. He worked as a professional musician for several years. In 1952 he enrolled at the Massachusetts Institute of Technology as a special student taking technical courses in preparation for research in electronic music. He spent 1954-1967 at the Massachusetts Institute of Technology continuing his research in this field as a Visiting Fellow in the Department of Humanities, as a Research Staff Member of the Research Laboratory of Electronics, and as a lecturer in the Department of Humanities where he taught in a program for interdisciplinary education. In 1967 he resigned from the Massachusetts Institute of Technology and established the firm of Ferretti-Lay, Inc., in Needham, Massachusetts, where he updated his research in computer music with emphasis on man-machine interaction. Professor Ferretti came to the University in August, 1970, as a Research Associate in Computer Science. In August, 1971, he was appointed as a Research Associate Professor.

Representative Publications:

Ferretti, Ercolino, 1965. The Computer as a Tool for the Creative Musician. In Computers for the Humanities A Record of the Conference Sponsored by Yale University, January 22-23, 1965. New Haven, Connecticut, Yale University Press, pp. 107-112.

Ferretti, Ercolino, 1968. Some Research Notes on Music with the Computer. American Society of University Composers Proceedings of the First Annual Conference, April 1966. pp. 38-41.

Boll, S.F., Ferretti, E. and Petersen, T., 1976. Improving Synthetic Speech Quality Using Binaural Reverberation. IEEE International Conference on Acoustics, Speech and Signal Processing April 12-14, 1976. Philadelphia, Pa. pp. 107-708.

MARTIN L. GRISS, Associate Professor of Computer Science; Ph.D., University of Illinois, 1971.

Professor Griss came to the University of Utah in 1973 and joined the Department of Computer Science in 1975. He has been involved with mathematical software and computational physics since 1963. He is currently interested in programming languages and software portability, with an emphasis on compiler and interpreter design. Since 1973 he has been a member of the Utah Computational Physics Group, working on computer algebra and symbol manipulation. He spent two postdoctoral years at California Institute of Technology working on large numerical programs in high energy physics phenomenology. Other areas of interest include sparse matrices, graph theory, and general numerical methods.

Representative Publications:

Griss, M.L., 1976. The Definition and Use of Data-Structures in REDUCE. Proc. of SYMSAC, 76. Ossining, New York.

Griss, M.L., 1976. The Algebraic Solution of Sparse Linear Systems via Minor Expansion. ACM Transactions on Mathematical Software.

Griss, M.L., 1977. Efficient Expression Evaluation in Sparse Minor Expansion Using Hashing and Deferred Evaluation. Proc. of 10th Hawaii International Conference on Systems Sciences.

ANTHONY C. HEARN, Ph.D., University of Cambridge, England, 1962. (on leave 1980-1981)

Professor Hearn is presently engaged in research concerned with the development and application of techniques necessary for the computer solution of algebraic problems in scientific research. This research is oriented toward both computer software development and the use of the developed programs in solving practical problems. Research activities include algebraic language standardization, algorithm development and the efficient modular division of such programs. In addition, work is in progress on the automatic generation of both algebraic and numerical programs by the use of symbolic computing techniques. A long-range objective of this work is the development of a completely automatic algebraic programming system which can be moved easily from computer to computer. Prior to his appointment at the University of Utah, he was Assistant Professor of Physics at Stanford University and before that was Senior Scientific Officer at the Rutherford High Energy Laboratory in England. Professor Hearn is presently a member of the Cambridge Philosophical Society, the American Physical Society, the Association for Computing Machinery (ACM), and the Society for Industrial and Applied Mathematics. He was an Alfred P. Sloan Fellow from 1967-1969.

Representative Publications:

Hearn, A.C., and Brown, W.S., Applications of Symbolic Algebraic Computation, Comp. Phys. Comm. 17 (1979) 207-215.

Hearn, A.C., Non-Modular Computation of Polynomial GCDs Using Trial Division, Proc. EUROSAM 79, published as Lecture Notes on Comp. Science, Springer-Verlag, Berlin, No. 72 (1979) 227-239

Hearn, A.C., and Norman, A.C., A One-Pass Prettyprinter, SIGPLAN Notices,

ACM, New York, 14, No. 12 (1979) 50-58

LEE A. HOLLAAR, Associate Professor of Computer Science; Ph.D., University of Illinois at Urbana-Champaign, 1975.

Before joining the faculty in 1980, Dr. Hollaar was an Assistant Professor of Computer Science and Senior Research Engineer for the Computing Services Office at the University of Illinois. His research interests are in systems engineering, primarily in the integration of conventional and novel hardware and software architectures. Specific areas have included advanced multi-sensor aircraft navigation systems, hierarchically-based energy management systems, and systems for the rapid retrieval of very large full-text databases (over 30 billion characters). The latter project, supported by the National Science Foundation, involves the design, analysis and simulation, and construction of specialized processors to efficiently implement functions normally performed by software (such as searching or combining index lists), along with the evaluation of the human interface (display formats, available commands, syntax, etc.) and its effect on the underlying hardware and software. He has also worked extensively with computerized photocomposition systems and operating systems design and implementation, recently for multiprocessor configurations.

Representative Publications:

Hollaar, L.A., 1978. Specialized Merge Processor Networks for Combining Sorted Lists. ACM Trans on Database Systems, 3,3:272-284.

Hollaar, L.A., 1979. A Design for a List Merging Network. IEEE Trans on Computers, C28,6:406-413.

Hollaar, L.A., 1979. Text Retrieval Computers. IEEE Computer, 12,3:40-50.

ROBERT M. KELLER, Associate Professor of Computer Science; Ph.D., University of California, Berkeley, 1970.

Professor Keller has general interests in Computer Science and Engineering. His more specialized area is Asynchronous Systems, their theories, and application of these theories to software and hardware systems which exploit and control concurrency. He has published in the areas of parallel computation, asynchronous modules, computer architecture, information retrieval, and program verifications. Examples include investigation of the concepts of maximal parallelism, the universality of asynchronous modules, verification techniques, and graphical formalisms. Current efforts are focused on techniques for exploiting concurrency and controlling resources in

multi-processing systems supporting applicative languages, the use of applicative languages for hardware specification and development, and the structure, specification, and proof of synchronization mechanisms in operating systems.

Representative Publications:

Keller, R.M., Lindstrom, G., and Patil, S., Data flow concepts for Hardware Design. Proc. IEEE Compcon (Feb. 1980).

Keller, R.M., Lindstrom, G., and Patil, S. A Loosely-coupled Applicative Multi-processing System. AFIPS Proc. 613-622 (June 1979).

Keller, R.M. Formal Verification of Parallel Programs. Communications of the ACM, 19, 7, 371-384 (July, 1976).

Keller, R.M. Towards a Theory of Universal Speed-independent Modules. IEEE Transactions on Computers, C-23, 1, 21-33 (January, 1974).

GARY E. LINDSTROM, Associate Professor of Computer Science; Ph.D., Carnegie-Mellon University, 1971.

Professor Lindstrom's principal research interests include programming language design, specification, and implementation; studies on control structures and their effect on programming style and execution efficiency; and programming aspects of parallel computer architectures. Prior to his appointment at the University of Utah, Professor Lindstrom was an Assistant Professor of Computer Science at the University of Pittsburgh. During the months of April-August 1975, Professor Lindstrom was a Visiting Scientist in the Department of Applied Math and Computer Science at the University of Grenoble, France, under NSF-CNRS United States-France Exchange of Scientists Program. He was also recipient of an NAS/NRC Travel Grant at Ljublijana, Yugoslavia in August 1971.

Representative Publications:

Lindstrom, G.E., Backtracking in a Generalized Control Setting, ACM Trans. on Programming Languages and Systems", 1,1 (July 1979) 8-26.

Lindstrom, G.E., 1978. Control Structure Aptness: a Case Study Using Top-Down Parsing. Proc. 3rd Int'l. Conf. on Software Engineering, Atlanta, pp. 5-12.

Lindstrom, G.E., Copying List Structures Using Bounded Workspace. Comm. ACM, 17, 4:198-202.

ELLIOTT I. ORGANICK, Professor of Computer Science; Ph.D., University of Michigan, 1950.

Professor Organick, who joined the faculty in 1971, is presently engaged in research in the organization of computer systems and in the teaching of programming linguistics. He is interested in the matching of computer and programming system structures and functions to those of the problems that people wish to solve. Other interests lie in development of new computer architectures for interpretive definable field machines and in reduction machines for reduction languages. He is engaged in research in various new system designs and has analyzed the structure of several major computer systems in three recent books. Other books and reports he has authored or co-authored have been texts for students and guides for teachers in basic concepts of programming and computer science. He has been active in professional work within the Association for Computing Machinery.

Representative Publications:

Organick, E.I., 1973. Computer System Organization: The B5700/B6700 Series. Academic Press, New York.

Organick, E.I., 1972. The Multics System: An Examination of its Structure. MIT Press.

Forsythe, A.I., Organick, E.I., Keenan, T.A., and Stenberg, W., 1975. Computer Science: A First Course, 2nd Edition. John Wiley.

SUHAS S. PATIL, Associate Professor of Computer Science; Sc.D., Massachusetts Institute of Technology, 1970.

Professor Patil is interested in theoretical and practical aspects of parallel processing computer systems. His research interests include effective computer organization for realizing computing systems from microcomputers, asynchronous programmable logic arrays, Petri nets, and theory of information pertinent to computing systems. Professor Patil was Associate Professor of Electrical Engineering and Computer Science at Massachusetts Institute of Technology before joining the Department of Computer Science in December 1975. He was a member of Project MAC at the Massachusetts Institute of Technology from 1966 to 1975 and served as Assistant Director of Project MAC from 1972 to 1974.

RICHARD F. RIESENFELD, Associate Professor of Computer Science , and Associate Chairman of Computer Science; Ph.D., Syracuse University, 1973.

Professor Riesenfeld received his B.A. in Mathematics from Princeton University, his M.A. in Mathematics from Syracuse University and his Ph.D. in Systems and Information Science from Syracuse University. He has been with the Department of Computer Science since August 1972. His doctoral dissertation involved developing mathematical schemes for computer aided geometric design of free-form curves and surfaces to be used with interactive computer graphics systems. This kind of work is sometimes called "computational geometry."

Representative Publications:

Barnhill, R.E., and Riesenfeld, R.F., (eds.), 1974. Computer Aided Geometric Design, Academic Press.

Lane, J.M., and Riesenfeld, R.F., A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, No. 2 (January 1980)

Cohen, E., Lyche, T., and Riesenfeld, R.F., An Application of Discrete B-Splines to Computer-Aided Geometric Design and Computer Graphics, Computer Graphics and Image Processing, (July 1980).

KENT F. SMITH, Research Associate Professor of Computer Science and Research Associate Professor of Electrical Engineering; M.S., Utah State University, Logan, 1958.

Professor Smith's interests lie in the design and use of integrated circuits and their application to computer systems. Prior to joining the University of Utah faculty, Professor Smith was responsible for integrated circuit design and testing at the Microcircuit Laboratory at the University of Utah Research Institute. Prior to that time he was the technical director for electrical engineering at General Instruments Advanced Microelectronics Lab. He holds a patent in circuits primarily concerning MOS and I2L integrated circuits.

Representative Publication:

Smith, K.F., 1971. One Cell Random Access Memory. Electronics, August 1971.

ROBERT E. STEPHENSON, Associate Dean, College of Engineering, Professor of Computer Science and Professor of Electrical Engineering; Ph.D., Purdue University 1952.

Professor Stephenson's principal interest lies in the use of computers as simulators of physical and discrete stochastic systems. He has published a textbook entitled, Computer Simulation for Engineers (Harcourt-Brace, 1971). Professor Stephenson joined the University of Utah faculty in 1946. During 1960-62 he was a Visiting Professor at Institut Technologi Bandung, Bandung, Indonesia. He was appointed Associate Dean of the College of Engineering in 1971.

THOMAS G. STOCKHAM, Jr., Professor of Computer Science (Resident at Soundstream, Inc.); Sc.D., Massachusetts Institute of Technology, 1959.

Professor Stockham joined the Computer Science faculty as Associate Professor in 1968, and was promoted to full Professor in 1970. He is currently performing research into the problems of using digital methods to process signals, with particular emphasis on the processing of pictures and sound. His research is aimed at exploring, developing and demonstrating specific techniques for problems such as automatic image deblurring, separation of voice from other voices and unwanted sounds, texture enhancement and classification, efficient high-quality speech compression, and image quality evaluation and control. He is researching development of advanced communications models for human vision and hearing, both of which play a vital role in this activity. Direction is consistently aimed toward the goal of making possible faster and more effective comprehension and use of important forms of sensory information via both human-assisted and automated means. Prior to his appointment at the University of Utah, he was a staff member of the Massachusetts Institute of Technology Lincoln Laboratories where he examined the uses of the principle of generalized superposition in processing auditory and visual signals. He has also been Assistant Professor of Electrical Engineering at Massachusetts Institute of Technology where he received the Goodwin Medal for excellence in teaching. He received the 1973 award for Outstanding Technical Achievement from the IEEE, Utah Chapter.

WILLIAM J. VIAVANT, Professor of Computer Science; Ph.D., University of Texas, 1954.

Professor Viavant's major interest is in the design and application of information processing systems for individual use, with emphasis on small independent systems which are highly interactive, using novel interfaces which better match human communication rates than traditional keyboards. Principal application areas are in individual instruction and in energy conservation. Other interests are interactive programming languages and the social and economic impact of computers.